(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 339 224 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89104268.1

(51) Int. Cl.4: G11C 7/00

(22) Date of filing: 10.03.89

(30) Priority: 29.04.88 US 187706

(43) Date of publication of application:
02.11.89 Bulletin 89/44

(84) Designated Contracting States:
DE FR GB

(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

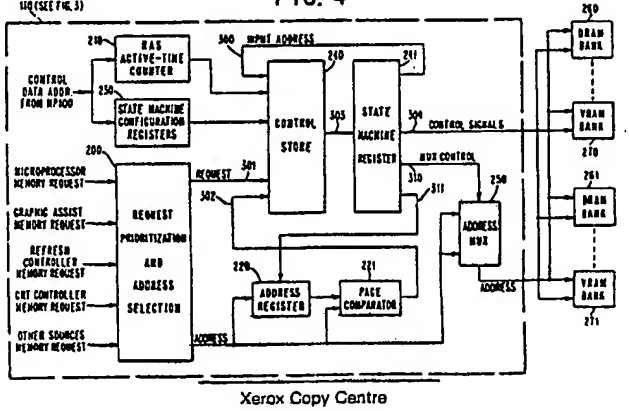
(72) Inventor: Bowater, Ronald John
Apeldoorn Whitenap Lane
Romsey Hampshire, S0518ST(GB)
Inventor: Larky, Steven Philip
225 West 83 Street Apt. 6-D
New York City, N.Y. 10024(US)
Inventor: St. Clair, Joe Christopher
2603 Valley View Cove
Round Rock Texas 78681(US)
Inventor: Sidoli, Paolo Gerardo
23 Halden Close Woodley
Romsey Hants S0517TS(GB)

(74) Representative: Rudolph, Wolfgang
IBM Deutschland GmbH Intellectual Property
Department Schönaicher Strasse 220
D-7030 Böblingen(DE)

(54) Memory controller.

(57) A flexible dynamic memory controller (110) that is operable with dynamic RAMS having a wide range of operating characteristics. These characteristics include different operating speeds for various memory functions, and the usage of memories (120 - 123). In a state machine, a special register (241) is utilized to control where in the sequence of operation, and for how long various delays must be inserted. The delays are dynamically determined by the memory controller (110) in accordance with the type of memory being accessed at a given time and the source of the request.

FIG. 4



MEMORY CONTROLLER

The invention is directed to a memory controller and a method for use with dynamic random access memories according to the preamble of claims 1 and 8 respectivly.

In order to control sequencing of dynamic random access memories (RAMs), a state machine is often used to provide the required flexibility. State machines are commonly used in large scale integrated (LSI) modules for controlling various functions on a cycle by cycle basis. They are used for memory controllers, bus controllers, simple processors, etc. Their value lies in the flexibility at design time to account for internal/external signals and conditions. Building in flexibility can be accomplished by testing external inputs or programming several sequences and choosing among them. Choosing the necessary sequences is not always possible though. For example, a memory controller may have to work with memory modules that are not available at the time the state machine is designed.

One way to allow flexibility in the module is to have the state machine receive its microinstructions from a memory array. The memory array is then loaded whenever the state machine state sequence needs to be changed. The array can be loaded while the module is in place in the final design. Unfortunately, this flexibility does not come for free. The memory array is often expensive in terms of chip area, and a means for loading the memory must be provided. In cases where a more limited flexibility is sufficient, a simpler solution is desired.

In memory controller state machines, for example, the amount of flexibility required is not very great -control over the length of time certain signals are active and the relationship between signals (RAS, CAS, etc.) is sufficient. Typically, a register contains a few bits describing which of several pre-defined paths should be selected to match the memory modules in use. In practice, just a few paths are defined, allowing for several speeds of RAMs (the "standard" speeds at the time the memory control part is introduced). This approach does not allow the controller to work most efficiently with faster memory modules.

There are a number of patents directed to memory controllers, each having certain advantages and disadvantages. A number of such patents are set forth below, none of which allows the selection of RAMs of different speeds at the same time.

U.S. Patent 4,691,289 to Thadem et al is directed to a video system controller which allows for the transfer of data between a display memory and a microprocessor that is used to control the controller and from the display memory to a CRT monitor. The transfer operations are controlled by the video system controller through a state machine that is configured with a plurality of standard cells connected in cascade arrangement, which can be configured as either a Moore or a Mealy state machine. Each state machine has a programmable logic array in which timing signals, when applied thereto, will cause a predetermined output to appear on the output on each of the standard cells. A logic means, depending upon whether the machine is a Moore type state machine or a Mealy type state machine logically manipulates the output of the programmable logic array to obtain the state output for that particular cell.

U.S. Patent 4,682,284 to Schrofer sets forth a memory subcontroller of a computer which includes a queue for storing read and write requests issued by memory using units to a memory apparatus for executing requests on the memory, and a circuit for administering the queue. When the queue is empty and the executing apparatus is ready to receive a request for execution, a request incoming from a using unit bypasses the queue; it is received by the executing apparatus directly and is not stored in the queue. Otherwise, the queue administration circuit stores the request in the queue and then awaits results of validity checks on the stored request. If the request is found to be invalid, generally the administration circuit discards the request from the queue by freeing the queue location or locations that store the invalid request to store the next received request. The invalid request is then overwritten by the next received request.

U.S. Patent 4,354,225 to Frieder et al describes a data processing system comprising an active and intelligent main store including a main memory, a main store controller for accessing the main memory in a manner allowing different address and data structures, and a main store bus connected to the controller. At least one processor of a first type is connected to the main store bus, this being an auxiliary processor for performing input-output and other operations. At least one processor of a second type also is connected to the main store bus, this being an execution processor for fetching, decoding and executing instructions.

All or some of either or both of the auxiliary processors and execution processors may be different. A supervisory processor for initiating configuring and monitoring the system is connected to the main store bus. A communication bus is connected to the processors of the first and second types and to the supervisory processor. A diagnostic bus connects the supervisory processor to each of the processors of the first and second types. An input-output bus ensemble is connected to the supervisory processor and to

each auxiliary processor. At least one device and associated device controller can be connected to the input-output bus ensemble. At least one direct memory access controller can be connected between the main store bus and the input-output bus ensemble. The limited flexibility required in certain designs is recognized. Rather than utilizing extra paths, extra states are included at critical points.

The object of the invention is to provide a flexible dynamic memory controller operable with dynamic RAMS having a wide range of operating characteristics is described, including different operating speeds for various memory functions, and the usage of memories.

The solution of this object is particularly described in the characterizing parts of claim 1, 2, 7, 8 and 11.

A register is provided in the state machine that selects where these extra states occur. In the simplest manifestation of the invention, each bit in the register selects whether or not a specific strategically placed state occurs in different sequences. In practice, this allows extra delays on signals, longer access time on memory modules, etc. In more complex cases, bits in the register indicate how long to remain in specific states. The delays are dynamically determined by the memory controller in accordance with the type of memory being accessed at a given time, and the source of the request.

In a state machine, a special register is utilized to control wherein the sequence of operation, and for how long various delays must be inserted. The delays are dynamically determined by the memory controller in accordance with the type of memory being accessed at a given time, and the source of the request.

The detailed description of the invention follows after the brief description of the drawings.

Fig. 1 is a state diagram showing programmable CAS access time, CAS precharge time and CAS start time for a memory controller;

Fig. 2 is a timing diagram indicative of the sequence of control signals generated by the memory controller of the invention;

Fig. 3 is a block diagram of a memory controller according to the invention;

Fig. 4 is a detailed block diagram of the flexible memory controller according to the invention;

Fig. 5 is a detailed block diagram of the page comparator set forth generally in Fig. 4;

Fig. 6 is a block diagram of two-way interleaved memory banks, including control signal connections; and

Figs. 7 - 10 are flow charts detailing the cycles generated by the flexible memory controller of Fig. 4.

It is the purpose of this invention to provide a highly flexible memory controller to control the sequencing of dynamic RAMs. A state machine is utilized which allows greater control over a large number of parameters. The choice of parameters is based on both the bank of RAM being accessed and the source of memory access. This is accomplished with a circuit that requires fewer gates than are normally required.

A flexible memory state machine is utilized as a dynamic memory controller to permit more flexibility than has been provided in available memory controllers. Memory attributes related to row address strobe (RAS) and column address strobe (CAS) have been made programmable. This allows the speed of memory access to be altered. The parameters that may be altered or modified include:

1. RAS precharge time -- time between active RAS is programmable.
2. CAS precharge time -- time between active CAS is programmable.
3. RAS access time -- time from when RAS first becomes active until valid read data is available or until write has been completed is programmable.
4. CAS access time -- time from when CAS first becomes active until valid read data is available or until write has been completed is programmable.
5. CAS start time -- time from the start of a memory cycle until CAS is made active is programmable (allowing extra data setup time for parity generation, data buffers, etc.).

Fig. 1 is a state diagram which illustrates programmable CAS access time, CAS precharge time, and CAS start time. The value of C in the circles represents the value of CAS for that state. In this figure C = 1 indicates CAS is inactive and C = 0 indicates CAS is active. The letters A, B, C and D represent individual bits in the state machine control register. If bit A is a ONE, then an extra CAS precharge state is added. Bit B allows for a delayed start of CAS. Bits C and D allow for a varied amount of CAS access time. Although bits A and B appear to have the same effect, the different entry points of the state machine (Entry 1 and Entry 2) determine their use. The state machine is entered at Entry 1 when in page mode, so CAS needs to be precharged. If, however, the memories were not in page mode, then the state machine is entered at Entry 2 and CAS precharge is not necessary, but it may still be necessary to delay the start of CAS active

time to allow for data setup time.

The invention also allows the "style" of memory access to be altered. Among the styles of memory that are supported:

1. Interleaved memory banks
2. Video memory
3. Fast page mode
4. Standard page mode
5. Static column mode
6. Nibble mode

The invention allows the speed parameters and style to be individually programmed based on two factors:

1. The bank of memory being accessed -- several different banks of memory can be controlled by the same memory controller (some could be video RAM, some could be static column mode, etc.), and the memory accesses are modified for each bank in order to maintain optimum performance.
2. The source of the memory request -- different sources of the request can also influence the access (such as requiring extra data hold time which effectively increases the CAS access time).

Fig. 2 is a timing diagram indicative of the sequence of control signals generated by the memory controller of the invention in accordance with the bank of memory accessed and the source of the memory request. The address signals, RAS signals, CAS signals and data signals are illustrated. How the cycle time of these control signals are dynamically modified to change the RAS access time and precharge time, and the CAS access time and precharge time as well as other control functions are described in detail below with reference to the following figures.

Referring to Fig. 2, Arrow 20 shows that the leading edge of CAS A active must occur after the Column address is output on Address A. Arrow 21 similarly shows how the leading edge of CAS B active must follow the Column address on Address B. Arrow 22 then shows that the subsequent leading edge of CAS A active must not occur until the address on Address A is set to the next value, in this case 1. The numbers in parenthesis represent the address as perceived to the requester. Since there are two memory banks being interleaved, memory address 0 in bank A corresponds to requester address 0, while in bank B memory address 0 is requester address 1. Similarly, memory address 1 in bank A corresponds to requester address 2, while in bank B memory address 1 is requester address 3. Arrows 23 thru 26 show the relationship between the leading edge of CAS active and memory data out. This is the CAS access time. Specifically, arrow 23 shows that the first leading edge of CAS A active plus the CAS access time yields DATA 0. Arrows 30 and 31 illustrate the time the leading edge of RAS active until data is available. This is the RAS Access time. Specifically, arrow 30 shows the RAS access time for bank A and arrow 31 shows the RAS access time for bank B. Arrows 32 and 33 illustrate the CAS precharge time, the time from CAS going inactive until CAS can be activated again. Arrow 40 illustrates the RAS precharge time, the time from RAS going inactive until RAS can be activated again. Arrow 41 illustrates the length of time RAS is active, which is controlled by the RAS active-time counter 210 in Fig. 4, which will be explained in detail later.

Refer now to Fig. 3 which illustrates a typical system that the memory controller is used in. The system includes a number of sources of memory requests 100-104, a flexible memory controller 110, and a number of banks of memory 120-123.

A microprocessor 100 generates memory requests in order to read and write memory as part of data update, instruction fetch, and other typical operations of a microprocessor. These requests are in the form of reads and writes, with a read request accompanied by its address, while a write request includes both the address and the data to be written at the address.

A graphic assist engine 101 generates memory requests when drawing lines, performing BitBlts, setting and reading pixels, fetching display lists, etc. These requests are a combination of reads, writes, and read-modify-writes.

A refresh controller 102 generates memory refresh requests. These are a special form of memory cycle wherein every DRAM/VRAM RAS line must be activated without a corresponding CAS cycle.

A CRT controller 103 generates video refresh requests. These are a special form of memory cycle wherein a special line (typically called TR/QE) is activated prior to the start of the memory cycle to denote an internal row transfer cycle. These requests are valid only for VRAM.

Other devices 104 can generate memory reads, writes, or special cycles dependent on the device. Typical examples are a DMA controller, virtual memory manager, or a bus controller honoring requests from another bus.

The flexible memory controller 110 recognizes requests from the various sources 100-104 and performs the desired memory operation. The memory controller 110 has the capability of controlling several types of dynamic random access memory and dual-port random access memory (typically called VRAM).

Memory modules 120-123 are arranged in banks. Each bank of memory contains several modules such that a bank is a complete word plus other bits optionally present for error detection and correction (EDC). A typical width of a word in present day systems is 32 bits with 4 more bits for parity or 6-8 bits for EDC. Memory banks may be of different sizes dependent on the density of the underlying memory modules. As an example, for a memory bank width of 32-bits, the bank sizes created with various density modules (ignoring parity and EDC) is as follows:

o	256 K byte bank of 8 - 256 Kbit modules (64k x 4-bit)
o	1 Mbyte bank of 32 - 256 Kbit modules (256K x 4-bit)
o	1 Mbyte bank of 8 - 1 Mbit modules (256K x 4-bit)
o	4 Mbyte bank of 32 - 1 Mbit modules (1M x 1-bit)
o	4 Mbyte bank of 8 - 4 Mbit modules (1M x 4-bit)
o	16 Mbyte bank of 32 - 4 Mbit modules (4M x 1-bit)

Dynamic memory modules have been optimized by manufacturers in order to provide the highest possible performance and bandwidth. One form of optimization relates to various methods of page mode (page mode encompasses standard page mode, fast page mode, nibble mode, and static column mode). A memory controller that can effectively use page mode is far more desirable and can provide a higher level of system performance.

In general, what page mode allows is faster subsequent access to memory locations on the same "page" as previous accesses. The size of the "page" varies based on the internal design and density of the memory module.

Fig. 4 is a detailed block diagram of the flexible memory controller 110 which functions as a sequential state machine. A request prioritization and address selector 200 receives all the memory requests from the various sources. The requests are prioritized by one of many schemes such as absolute fixed priority (requests are always ordered in the same manner from most urgent to least, rotating requests are given priority based on how long since last honored), age (requests are given priority based on how long the request has been active), or some combination of the above. What is relevant is that one and only one request (along with its address) passes through the prioritization section.

A control store 240 which contains microcode describing the sequences the memory controller can perform. The control store may be any of a variety of technologies depending on the manner in which the flexible memory controller is realized. If the flexible memory controller is to be contained in a semi-custom chip such as a gate array, it may be composed of a plurality of logic gates such that the resulting logic gates match the desired input/output function of the control store. These gates may be the result of a logic reduction on the control store input/output function or its truth table. The control store can also be implemented in ROM (read only memory) on a semi-custom chip. If the flexible memory controller is implemented in a plurality of chips, then the control store may be composed of any of a number of commercially available ROMSs and PROMs. The control store, based on an input address 300 coupled with the contents of state machine configuration registers 230 and various other inputs such as the requests 301, RAS down counter status 210, and page comparator 221 out on 302, generates output signals 303 that are then latched in state machine registers 241. RAS active time counter 210 and registers 230 receive control, data and address information (for loading their internal registers) from microprocessor 110. Details of the functions of these devices is described shortly.

The output of the state machine registers 241 can be divided into three general categories: (1) feedback information on line 300 which are necessary to determine the next address for the control store 240, (2) memory control signals 304 for the memory banks, and (3) internal memory controller control signals 310 and 311 which are applied to multiplexor 250 and address register 220, respectively. It should be noted that memory control signals may be processed by conditioning logic before connecting to the memory modules.

The previous address register 220 and page comparator 221 operate in concert to determine if the current memory request can be treated as a page-mode memory cycle. Fig. 5 is a detailed description of the page comparator 221. The page comparator 221 compares the bank and row address portion of the previous address stored in the address register 220 with the bank and row address portion of the current address from prioritization and address selection logic 200. The previous address was latched in the address register 220 at the beginning of the previous cycle, indicated on line 311. The row address portion

varies depending on the size of the memory bank and whether or not banks are being interleaved. For example refer to Table 1 below:

TABLE 1

Bank Type	Row	Column
256K by n	17:9	8:0
1M by n	19:10	9:0
4M by n	21:11	10:0
Interleaved 256K by n	18:10	9:1
Interleaved 1M by n	20:11	10:1
Interleaved 4M by n	22:12	11:1

If both the bank and row address portion match, then a page mode operation may occur. A top bits comparator 500 performs the comparison for all the bits above the page address plus bit 12 since they are common to all banks as either bank address bits or row address bits. Depending on the bank's memory modules, address bits 9, 10, and 11 must also match. EXCLUSIVE-NOR gates 510, 511, 512 output a ONE if the respective input address bits match, and a ZERO if there is no match. The output of these EXCLUSIVE-NOR gates are OR'd with the signals ignore address <11>, <10>, and <9>. The resulting signals are all AND'd together with AND gates 530 and 540 to determine if there is page-mode address match. Determining the values of ignore address <11>, <10>, and <9> may be a complicated matter dependent upon how many banks of memory there are and what kind of different modules are used (high order address bits need to be looked up in a table of some kind for this determination). However, careful examination reveals that by comparing ALL the top bits, the old values of ignore address <11>, <10>, and <9> may be used. If the top address bits match, then also the correct bits, 11, 10, and 9 are checked for a match. If the top address bits don't match, then the comparator 500 will prevent a page-mode cycle. This allows a very fast comparison to be used without need for determining the "pro per" row address.

Returning to Fig. 4, the RAS active-time counter 210 prevents the memory controller from exceeding the maximum RAS active-time specification. Typically dynamic memory modules specifications allow a limited duration of RAS active (page-mode operation) time. It is necessary to keep a count of how long RAS has been active in order to artificially bring the DRAM out of page-mode in the presence of a long string of page mode addresses. The counter is incremented every system clock cycle and compared to a preset value which is determined under control of the signal inputs from microprocessor 110. The maximum count is based on the speed of the system clock and the modules specification of maximum RAS active time (typically 10 microseconds). For a 24 MHz system clock, this translates into a maximum count of 240. Because the maximum count could be reached during a memory cycle and RAS could not be deactivated immediately, the preset value is set lower than the number of clocks within the spec to allow for any memory operation to be completed. Continuing the previous example, if the longest memory cycle is 640 ns, then the maximum count is set to 224 - (10,000 ns - 640 ns/(1/24 MHz)) rounded down.

The address multiplexor 250 allows the current address to be multiplexed into a row portion and a column portion. The multiplexor is 2N-way, with N equaling the number of different row/column address portions the various banks require.

Control signals 304 include a separate RAS for each bank of RAM (260, 261, 270, 271), a set of CAS corresponding to the number of banks that can be interleaved (a set is composed of a separate CAS for each byte that can be individually written), write enable lines, TR/QE (for VRAM) lines, and output enable lines.

The state machine configuration registers 230 provide inputs that can be tested for various conditions that can then modify the type of memory cycle to be performed. Changing the contents of the state machine configuration registers provides programmable memory cycles. The registers 230 respond to the signal inputs from microprocessor 100 which are indicative of the source of the memory request and the type of memory bank accessed, to change the contents of the registers. The control store 240 responds to the contents of the register 230 to change the duration of the cycle time of the control signals applied to the selected memory bank. Specifically the access time and precharge time of RAS and CAS are modified. This is described in detail relative to the flow charts of Figs. 7-10.

The memory banks 260, 261, 270, and 271 can be two-way interleaved, such that banks 260 and 261 act as a pair, as do 270 and 271. Fig. 6 shows how control signals connect to banks of memory for two-way interleaving. These control signals are illustrated in Fig. 2. Each bank of memory has its own RAS line, there are (in this example) 4 CAS lines for each set of banks being interleaved (assuming byte selection for a 32-bit word), there are separate address and other control signals (WE, OE, etc.) for each set of banks being interleaved. Alternate addresses (on a word boundary) are stored in alternate banks, allowing faster access to a contiguous chunk of memory because access to the banks can be overlapped. An additional benefit of interleaving is that the page size is effectively doubled because now there is an active row in both banks. The page comparator must take this into account when comparing previous and current addresses. The flexible memory controller allows either DRAM interleaving, VRAM interleaving, both, or neither. In a typical system configuration, there will be more than one pair of memory banks, all of which can be interleaved. It is also possible to interleave more than two-way, however, current technology provides little benefit for the extra cost of separate CAS signals and internal control.

Refer now to Figs. 7 - 10 which illustrate the flowchart of the states the flexible dynamic memory controller can assume as manifested at the output of the state machine register 241 of Fig. 4. The changing of states is assumed to be based on an input clock to the state machine. This clock has a period (1/frequency) which determines how long a signal that is activated in one state and deactivated in a later state stays active.

The initial state diagram flow is as set forth below. The initial state upon reset is the IDLE state 500 shown in Fig. 7. If a "regular" memory cycle request is detected (i.e. it flows through the request prioritization logic 200 of Fig. 4) then State 501 is entered, if a refresh request is detected then State 513 of Fig. 10 is entered via line 570. Other request types (such as a VRAM serial refresh request from CRT Controller) also branch off of State 500. When in State 501, the request address is decoded in order to determine which, and what type, bank is being accessed. If the bank being accessed is interleaved, then State 502 is entered, otherwise State 503 is entered. In State 502 the non-selected bank's address is enabled (i.e. the odd bank in the case of an even bank request). The odd and even bank address buses, address A and address B lines of Fig. 6, must both have been enabled in the case of interleaved banks because the Row Address Strobes for both odd and even banks are activated together. Subsequent page mode cycles can then proceed in either the odd or the even bank. The address buses are not enabled together in order to minimize the number of simultaneously switched output pins. By enabling the address buses separately, higher power drivers can be utilized. In State 503 the selected banks' address is enabled on either address A or address B line of Fig. 5 and then State 504 is entered. In State 504 the Row Address Strobe (RAS) is activated for both odd and even banks, RAS A and RAS B of Fig. 2, if interleaved address or just the selected bank (odd or even) if the address is for a noninterleaved bank. RAS stays activated until state 512 is entered as shown in Fig. 9. If the clock period of the state machine is less than the row address hold time then state 506 follows state 504, otherwise state 507 (Fig. 7) follows state 504. While in State 506, no signals are changed, this allows extra time for the Row Address to be held after the falling edge of RAS, a parameter that is commonly specified in dynamic memory specifications.

As shown in Fig. 8, in state 507 the address is changed from the multiplexed row address to the column address. This refers to Fig. 4, wherein the address mux 250 is changed via the mux control 310, as specified by the type and size of memory bank as shown in Table 1 (bank type versus row and column address). In State 508, the Column Address Strobe (CAS) is activated. CAS remains activated until either State 511 (Fig. 9) or State 53n is reached.

The control of CAS access time is as set forth below. If the clock period of the state machine is less then the CAS access time then State 52n, that is, State 521 or 522, follows State 508, otherwise State 509 follows state 508. While in State 52n, no signals are changed, this allows extra time for CAS access before the memory controller signals it has completed the memory cycle. The n in State 52n is the number of extra clock periods of access time that is needed. That is, if 1 extra clock cycle is needed, State 521 is entered, and if 2 extra clock cycles are needed, State 522 is entered. State 521 always follows State 522, and after State 521, State 509 is entered. If more than 2 extra CAS access states are required, State 523 is added ahead of State 522, State 524 is added ahead of 523, etc. CAS access time is commonly specified in dynamic memory specifications. As previously set forth relative to Fig. 4, State machine configuration registers 230 provide the inputs to control store 240 for determining CAS access time dynamically as a function of the source of memory request and the type of memory bank being accessed.

In State 509 CAS is held active and the memory controller returns DONE to the requester - in the case of a write the requester no longer needs to assert the data lines and in the case of a read the requester can now latch the memory data. In State 511 CAS is held active if the active bank style is static column mode memory. Static column mode memory does not require CAS be brought inactive between page mode

cycles. For static column mode memory banks, States 53n, that is States 531 and 532, are never entered.

Control of CAS precharge time is now set forth. Again, as previously set forth, CAS precharge time is determined dynamically by control store 240 in response to input signals from registers 230 which are indicative of the source of the memory request and the type of memory bank being accessed. Assuming a page mode or fast-page mode memory bank is being accessed, there is a minimum CAS precharge time specification that must be met. If the minimum CAS precharge time is longer than the clock period then extra CAS precharge states need to be added. The default CAS precharge time is the time spent in one state, State 511. If extra CAS precharge time is required then State 53n (where n is the number of extra clock periods of CAS precharge time needed) follows State 509, otherwise State 511 follows State 509. That is, if 1 extra clock cycle is needed, State 531 is entered, and if 2 extra clock cycles are needed State 532 is entered. State 531 always follows State 532, and after State 531, State 511 is entered. If more than 2 extra CAS precharge states are required, State 533 is added ahead of State 532, State 534 is added ahead of 533, etc.

Page Mode Idle is as follows. In State 511 several tests are performed every cycle. The highest priority test is the RAS active time count (from the RAS active time counter 210 of Fig. 4), if this count exceeds the programmed value then State 512, RAS Precharge, is entered. RAS access time and RAS precharge time, as previously set forth, are determined by control store 240 in accordance with input signals from registers 230 which are indicative of the source of the memory request and the type of memory bank being accessed. The second highest priority test is whether or not a request has occurred. If a request has not occurred, then the state machine stays in State 511. If, however, a request has come along, then the page comparator 221 (Fig. 4) is checked to see if the memory can stay in page mode. If there is a page hit, then a test for delay of CAS start must be made to determine if State 508, CAS Cycle Start follows State 511 or if State 55n follows. If there is a request but no page hit, then State 512, RAS Precharge, is entered.

Control of delay CAS start is as follows. Some requesters require a delay of the CAS start in order to allow the requesting master extra time to generate data or parity when the request is a write. If the master needs extra time and State 508 (Fig. 8) would have normally followed State 511, then State 55n (where n is the number of extra clock periods required before the master is ready), follows instead. That is, if 1 extra clock cycle is required State 551 is entered and if 2 extra clock cycles are required State 552 is entered. State 551 always follows State 552, and after State 551, State 508 is entered. If more than 2 extra delay states are required, State 553 is added ahead of State 552, State 554 is added ahead of 553, etc.

Control of RAS precharge time is as follows. If the minimum RAS precharge time is longer than three clock periods (the time in State 512, 501, and 503) then extra RAS precharge states need to be added. If extra RAS precharge time is required then State 54n (where n is the number of extra clock periods of precharge time needed) follows State 512, otherwise either State 500 (no request pending) or State 501 (request pending) follows State 512. That is, if 1 extra clock cycle is required State 541 is entered and if 2 extra clock cycles are required State 542 is entered. State 541 always follows State 542, and after State 541, State 500 or State 501 is entered (again, depending on a pending request). If more than 2 extra RAS precharge states are required, State 543 is added ahead of State 542, State 544 is added ahead of 543, etc.

Referring to Fig. 10, control of refresh RAS active time is as follows. Whenever a refresh request is detected State 513 is entered after the previous cycle is completed (State 541 or State 512 of Fig. 9) or from IDLE. In State 513, the refresh address is enabled onto the odd address bus and then State 514 is entered. In State 514, the refresh address is enabled onto the even address bus and then State 515 is entered. In State 515 RAS is activated to all banks of memory. If the clock period of the state machine is less than the minimum time then State 56n, where n is the number of extra clock periods of RAS active time needed, follows state 515, otherwise State 512 follows State 515. That is, if 1 extra clock cycle is required State 561 is entered, if 2 extra clock cycles are required State 562 is entered, and if 3 extra clock cycles are required Sate 563 is entered. While in State 56n, no signals are changed, this allows extra time for refresh RAS time before RAS is deactivated. State 562 always follows State 563, State 561 always follows State 562, and after State 561, State 512 is entered.

If more than 3 extra RAS active states are required, State 564 is added ahead of State 563, State 565 is added ahead of 564, etc. Refresh minimum RAS time is commonly specified in dynamic memory specifications and is usually the same as the RAS access time.

Other memory control signals are also generated, as now described. The above description detailed the changing of signals that are sequenced in a unique manner by the flexible memory controller, responsive to signal inputs from registers 230. Other signals, such as write enables and output enables also need to be sequenced by the memory controller. The sequencing of these other signals is not detailed, as the correct sequencing is a well known and documented in many dynamic memory specifications. However, if other signals are sequenced in a manner similar to the Column Address Strobe (again, such as write enable),

then the number of CAS cycles (States 521, 522, 52n of Fig. 7) could be increased in order to gain longer write enable overlap time.

Generation of n for State 5xn selection is as set forth below. The preceding analysis of the state flowchart contained several references to a value n where n is the extra RAS precharge time or the extra CAS access time, etc. The state machine configuration registers 230 (Fig. 4) contain the relevant n value for each desired attribute. The following Table 2 shows example attribute values.

The Table 2 can be stored in a series of registers comprising registers 230.

TABLE 2

BANK SPECIFIC PARAMETERS					
Parameter	States	Bank A	Bank B	Bank C	Bank D
CAS Access Time	52n	0	0	1	0
CAS Precharge Time	53n	0	0	1	1
RAS Precharge Time	54n	1	1	2	2
RAS Access Time	56n	2	2	3	3
Row Address Hold Time	506	no	no	yes	no
Static Column Mode		no	no	yes	no
Interleaved		yes	yes	no	no
REQUESTING MASTER PARAMETERS					
Parameter	States	Microprocessor	Graphic Assist Engine	Refresh Controller	CRT Controller
CAS Delay Time	55n	1	0	0	0

The following example state sequences use the state number of Figs. 7 - 10 along with the attribute definitions of the above Table 2.

The sequence of states for a microprocessor requesting access to Bank A, based on the above table is as follows: State 500, State 501, State 502 (since A is interleaved), State 503, State 504, State 507, State 508, State 509, State 511, and then after no request and a timeout - State 512, State 541, State 500.

The sequence of states for a microprocessor requesting access to Bank C and then an access to C on a different page, based on the above table is as follows: State 500, State 501, State 503, State 504, State 506, State 507, State 508, State 521, State 509, State 531, State 511, State 512, State 542, State 541, State 501, State 503, State 504, State 506, State 507, State 508, State 521, State 509, State 531, State 511, and then after no request and a timeout - State 512, State 542, State 541, State 500.

The sequence of states for the refresh controller request, based on the above table is as follows: State 500, State 513, State 514, State 515, State 563 (longest of RAS access is chosen), State 562, State 561, State 512, State 542, State 541, State 500.

In summary, a memory controller has been described which responds to memory access requests from a plurality of different processors to provide control signals to a plurality of memory banks which may have different cycle times of operation. The duration of the cycle time of the control signals is dynamically determined as a function of the source of the memory request and the type of memory bank accessed.

Industrial Applicability

It is an object of the invention to provide an improved memory controller.

It is another object of the invention to provide an improved memory controller in which the speed of memory control sequencing is determined in part by the bank of memory being accessed.

It is yet another object of the invention to provide an improved memory controller in which the speed of the memory control sequencing is determined in part by the source of the memory access.

It is still another object of the invention to provide an improved memory controller in which the speed of

memory control sequencing is determined in part by the style of memory being accessed.

It is yet still another object of the invention to provide a dynamic memory controller in which the length of RAS precharge time, CAS precharge time, RAS access time and CAS start time are independently programmable to allow support of many different types and speeds of dynamic RAMs.

It is a further object of the invention to provide a dynamic memory controller in which the speed of memory control sequencing can be modified to match the system clock of the memory controller module.

It is yet a further object of the invention to provide a dynamic memory controller in which a state machine control register selects the inclusion of various states on an individual state basis.

Claims

1. Method of operating a computing system which include at least two devices capable of making a plurality of different types of memory access requests to a plurality of memory banks, with some of said memory banks having a different speed of operation, with a memory controller responsive to said memory access requests for providing control signals to said memory banks which are indicative of sequences of operation of the memory banks, characterized by
decoding a memory access request by the memory controller (110) to determine which one of the plurality of memory banks (120-123) is to be addressed; and
providing said control signals from the memory controller to said one memory bank, with the duration of the cycle time of said control signals being dynamically determined as a function of one of the type of said one memory bank, and the source of the memory request.

2. Method of operating a computing system which includes a plurality of processors which are capable of making a plurality of different types of memory access requests to a plurality of different types of memory banks, with some of said memory bank having a different cycle time of operation, with a memory controller responsive to said memory access requests for provision of control signals to said memory banks which are indicative of sequences of operation of said memory banks,
characterized by
decoding a memory access request by the memory controller (110) to determine the row and column address of the selected memory bank (120 - 123) to be addressed;
providing a row address strobe signal (RAS) and a column address strobe signal (CAS) by the memory controller to said selected memory bank for controlling the sequence of operation of said selected memory bank; and
determining dynamically by said memory controller the row address strobe (RAS) precharge time and column address strobe (CAS) precharge time of said row address strobe signal (RAS) and column address strobe signal (CAS) as a function of one of the type of memory bank being accessed and the source of the memory access request.

3. Method of claim 2,
characterized by
determining the maximum time of a row address strobe (RAS) is active based on a maximum predetermined count which is a function of the system clock time.

4. Method of a claim 2,
characterized by
determining if the memory access request is a page mode request, and
determining if the following memory access request is for the same page as the previous request, and if so concurrently processing such requests.

5. Method of claim 2,
characterized by
determining a row address strobe (RAS) access time and a column address strobe (CAS) access time of said row address strobe signal (RAS) and said column address strobe signal (CAS) as a function of one of the type of memory bank being accessed and the source of the memory access request.

6. Method of operating a computing system including a plurality of processors which are capable of making a plurality of different types of memory access requests to a plurality of memory banks which may have different cycle times, further including a memory controller responsive to said memory access requests for providing sequences of control signals to said memory banks which are indicative of sequences of operation for the memory banks,
characterized by
decoding a memory access request from a given one of the processors (100) by said memory controller

(100) to determine the row and column and bank address of the selected memory bank (120 - 123);
providing a row address strobe signal (RAS) by said memory controller;
providing a column address strobe signal (CAS) by said memory controller;
determining dynamically by said memory controller if extra column address strobe (CAS) access time for said column address strobe signal (CAS) is required as determined by one of the type of memory bank selected and the source of the memory access request;
determining dynamically by said memory controller if extra column address strobe (CAS) precharge time for said column address strobe signal (CAS) is required as determined by one of the type of memory bank selected and the source of the memory access request;
determining dynamically by said memory controller if extra row address strobe (RAS) precharge time for said row address strobe signal (RAS) is required as determined by one of the type of memory bank selected and the source of the memory access request;
providing said row address strobe signal (RAS) with the determined precharge time to the selected memory bank;
providing said a column address strobe signal (CAS) with the determined access time and precharge time to the selected memory bank; and
providing said bank address signal to the selected memory bank, such that said selected memory bank may perform the sequences of operation controlled by said row address strobe signal (RAS) and said column address strobe signal (CAS).

7. Method of claims 1 to 6,
characterized by
memory access requests from at least two of said processors for provision of control signals to at least two of said memory banks which are indicative of different sequences of operation of said two memory banks, said method comprising the steps of:
decoding said concurrent memory access requests from said two processors by said memory controller to determine the row and column address of each of the two selected memory banks to be addressed;
providing a first and second row address strobe signal (RAS) and a first and second column address strobe signal (CAS) by said memory controller to each of the first and second selected memory banks, respectively, for controlling the sequence of operation of said first and second selected memory banks; and
determining dynamically by said memory controller the different row address strobe (RAS) precharge time and column address strobe (CAS) precharge time of said first and second row address strobe signals (RAS), and said first and second column address strobe signals (CAS) as a function of one of the type of said first and second memory banks, respectively, being accessed and the source of the memory access request from the first and second processors, respectively.

8. Computing system, including
at least two devices capable of making a plurality of different types of memory access requests,
a plurality of memory banks which may have different speeds of operation,
a memory controller,
characterized by
means (200) for decoding a memory access request from said at least two devices to determine which one of said memory banks (120 - 123)_is being addressed, and
means (241) for providing control signals to the addressed memory bank (120 - 123), with the duration of the cycle time of said control signals being determined dynamically as a function of one of the speed of the addressed memory bank, and the source of the memory access request.

9. Computing system, as set forth in claim 8 for operating with a method of one of claims 1 to 7, characterized by
means for decoding a memory access request from one of the processors to determine which one of said memory banks is being addressed;
means for providing a row address strobe signal (RAS) and column address strobe signal (CAS) to the memory bank being addressed to control the sequence of operation of the memory bank; and
means for dynamically determining the row address strobe (RAS) precharge time and the column address strobe (CAS) precharge time of said row address strobe signal (RAS) and said column address strobe signal (CAS), respectively, as a function of one of the type of memory bank being accessed and the source of the memory access request.

10. Computing System as set forth in claim 8 or 9, characterized by
a counter (210) which is incremented every clock cycle and compared to a maximum count based on the system clock for determining the maximum time row address strobe (RAS) is active.

11. Computing system as set forth in claims 8 or 9 or 10,
characterized by
means (200) for decoding a memory access request to a selected memory bank from a given one of said processors to determine the bank, row and column address of said given one of said processors,
means for providing a row address strobe signal (RAS);
means for providing a column address strobe signal (CAS);
means for dynamically determining if extra column address strobe (CAS) access time for said column address strobe signal (CAS) is required as determined by one of the type of memory bank selected and the source of the memory access request;
means for dynamically determining if extra column address strobe (CAS) precharge time for said column address strobe signal (CAS) signal is required as determined by one of the type of memory bank selected and the source of the memory access request;
means for determining if extra row address strobe (RAS) precharge time for said row address strobe signal (RAS) is required as determined by one of the type of memory bank selected and the source of the memory access request;
means for providing said row address strobe signal (RAS) with the determined precharge time to the selected memory bank;
means for providing said column address strobe signal (CAS) with the determined access time and precharge time to the selected memory bank; and
means for providing the decoded bank, row and column address signals to the selected memory bank, for enabling the selected memory bank to perform the sequences of operations controlled by said row address strobe signal (RAS) and said column address strobe signal (CAS).

12. Computing system, with
a plurality of processors capable of making a plurality of different types of memory access requests; a plurality of memory banks which may have different cycle times of operation,
characterized by
means (200) for decoding on a prioritization basis a memory access request for a selected memory bank from a given one of said processors, for accessing the selected memory bank by said given one of said processors;
means (304, 310) for providing input signals to said memory controller which are indicative of the type of memory bank selected and the source of the memory access request for modifying the cycle time of said control signals.

13. Computing system, as set forth in one of claims 8 to 12,
characterized by
means for dynamically determining the different row address strobe (RAS) precharge time and the column address strobe (CAS) precharge time of said first and second row address strobe signals (RAS) and said first and second column address strobe signals (CAS),
respectively, as a function of one of the type of said first and second memory banks, respectively, being accessed and the source of the memory access request from the first and second processors, respectively, and
means for determining the different row address strobe (RAS) access time and a column address strobe (CAS) access time for said first and second row address strobe signals (RAS) and said first and second column address strobe signals (CAS), respectively, as a function of one of the type of said first and second memory banks, respectively, being accessed and the source of the memory access request from the first and second processors, respectively.

FIG. 1

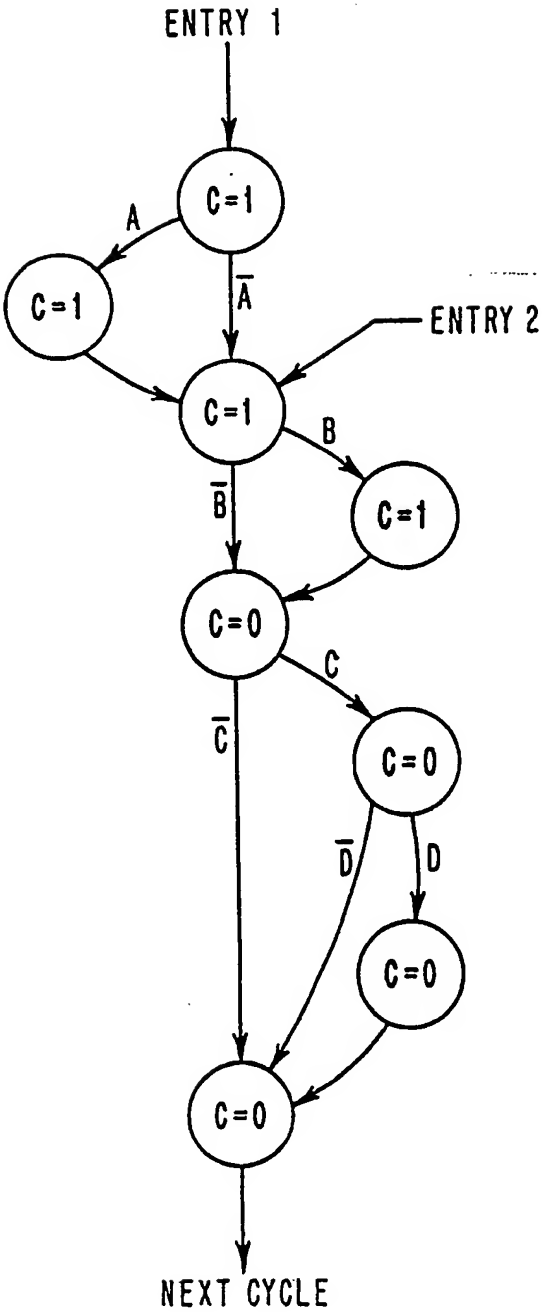

ENTRY 1
C=1
A
C=1
$\overline{A}$
ENTRY 2
C=1
B
$\overline{B}$
C=1
C=0
C
$\overline{C}$
C=0
$\overline{D}$
D
C=0
C=0
NEXT CYCLE

FIG. 2

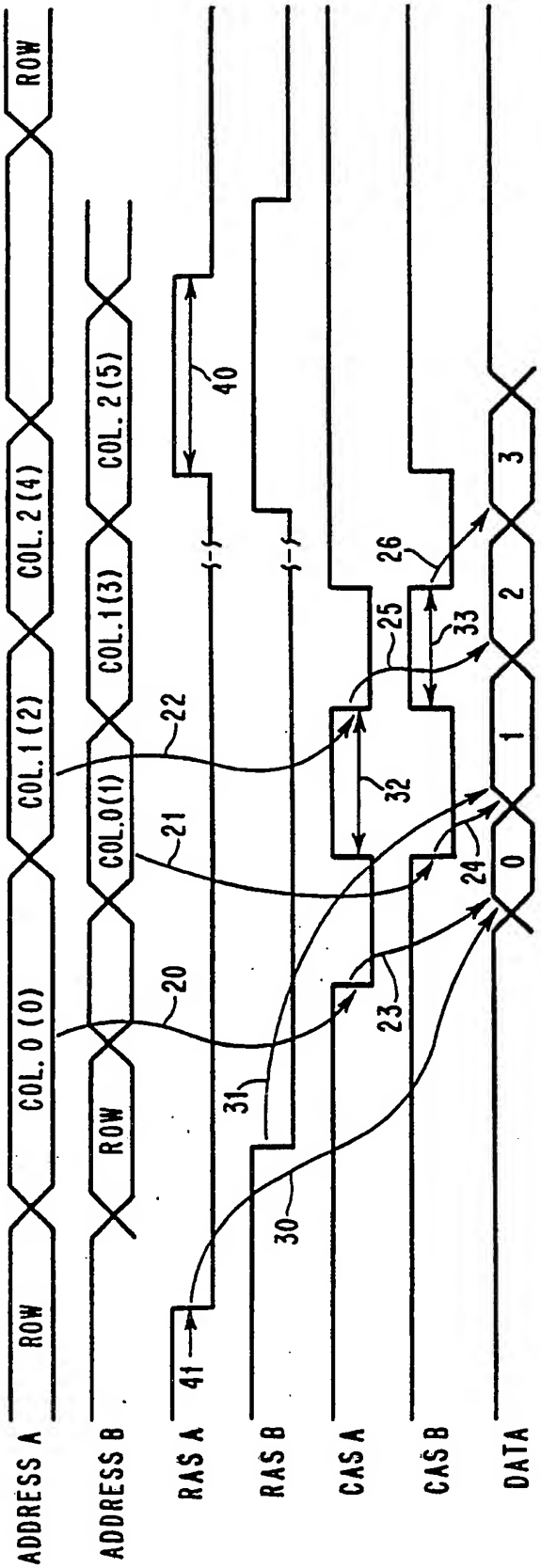

ADDRESS A
ADDRESS B
RAS A
RAS B
CAS A
CAS B
DATA
ROW
COL. 0 (0)
COL. 1 (2)
COL. 2 (4)
ROW
COL. 0 (1)
COL. 1 (3)
COL. 2 (5)
0
1
2
3
20
21
22
23
24
25
26
30
31
32
33
40
41

FIG. 3

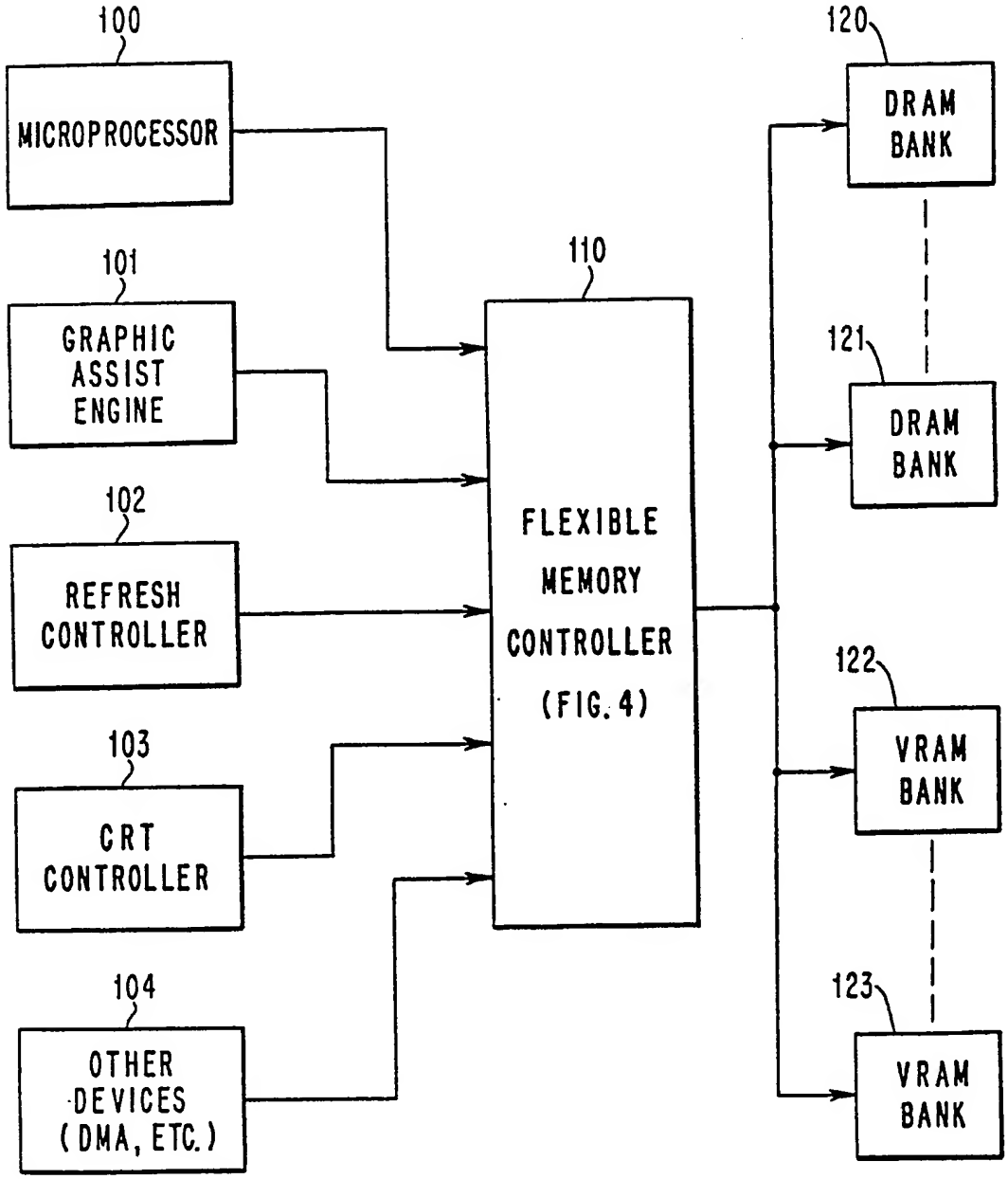

100
MICROPROCESSOR
101
GRAPHIC ASSIST ENGINE
102
REFRESH CONTROLLER
103
CRT CONTROLLER
104
OTHER DEVICES (DMA, ETC.)
110
FLEXIBLE MEMORY CONTROLLER (FIG. 4)
120
DRAM BANK
121
DRAM BANK
122
VRAM BANK
123
VRAM BANK

FIG. 4

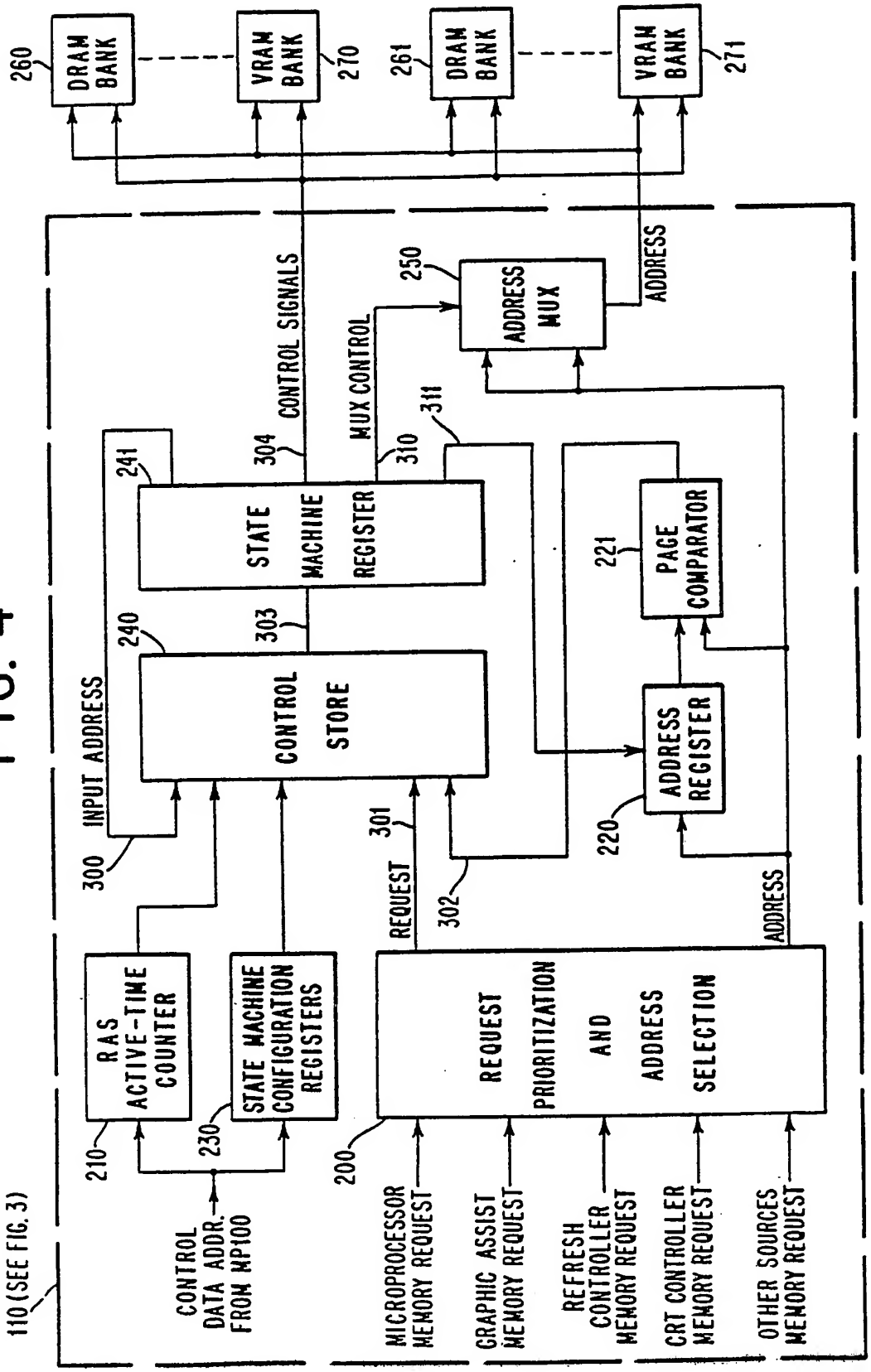

260
DRAM BANK
VRAM BANK
270
261
DRAM BANK
VRAM BANK
271
CONTROL SIGNALS
304
250
ADDRESS MUX
ADDRESS
MUX CONTROL
310
311
241
STATE MACHINE REGISTER
303
240
CONTROL STORE
INPUT ADDRESS
300
221
PAGE COMPARATOR
220
ADDRESS REGISTER
301
REQUEST
302
ADDRESS
RAS ACTIVE-TIME COUNTER
STATE MACHINE CONFIGURATION REGISTERS
210
230
200
REQUEST PRIORITIZATION AND ADDRESS SELECTION
110 (SEE FIG. 3)
CONTROL DATA ADDR. FROM MP100
MICROPROCESSOR MEMORY REQUEST
GRAPHIC ASSIST MEMORY REQUEST
REFRESH CONTROLLER MEMORY REQUEST
CRT CONTROLLER MEMORY REQUEST
OTHER SOURCES MEMORY REQUEST

FIG. 5

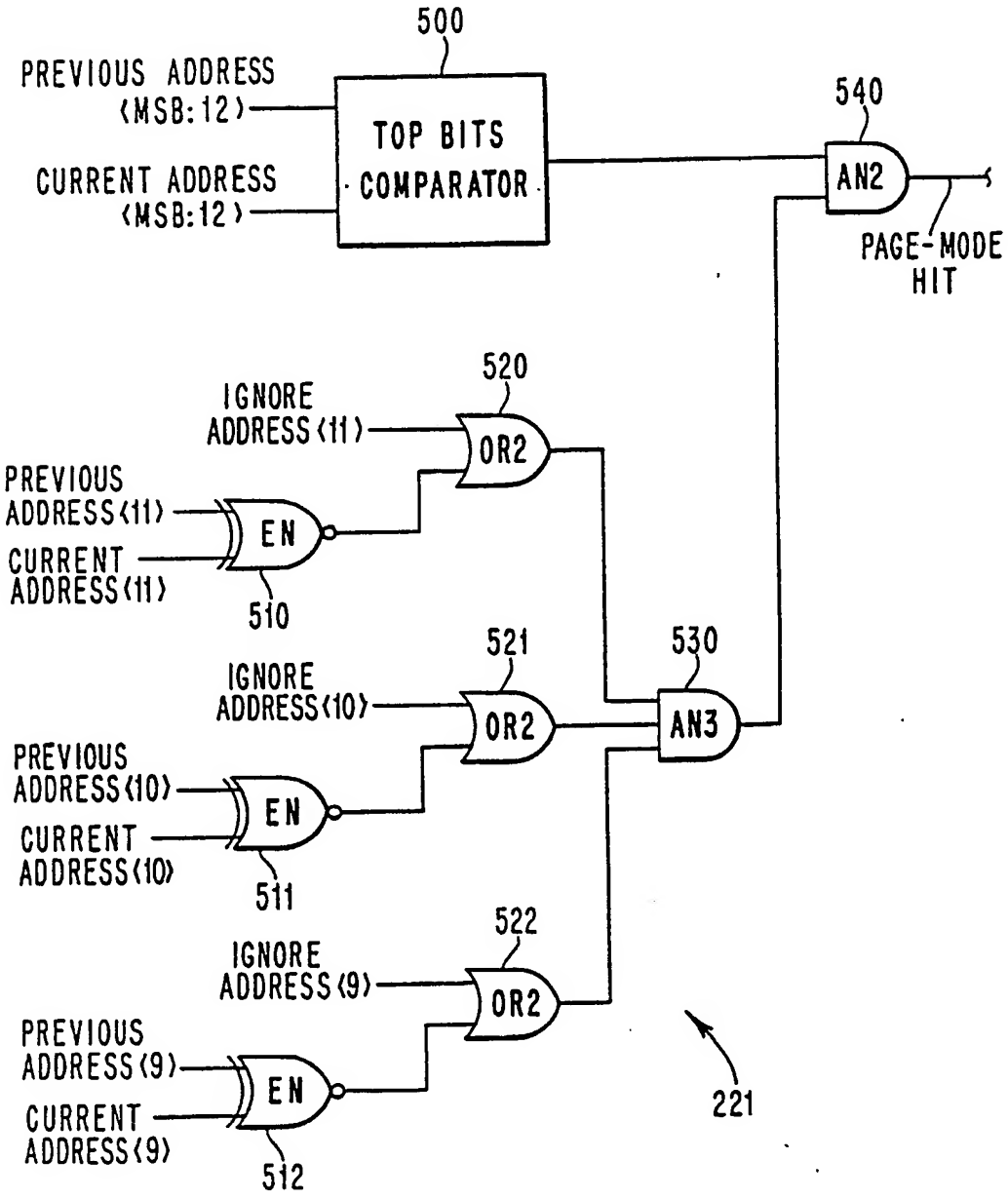

500
PREVIOUS ADDRESS (MSB:12)
CURRENT ADDRESS (MSB:12)
TOP BITS COMPARATOR
540
AN2
PAGE-MODE HIT
520
IGNORE ADDRESS(11)
OR2
PREVIOUS ADDRESS(11)
CURRENT ADDRESS(11)
EN
510
521
530
IGNORE ADDRESS(10)
OR2
AN3
PREVIOUS ADDRESS(10)
CURRENT ADDRESS(10)
EN
511
522
IGNORE ADDRESS(9)
OR2
PREVIOUS ADDRESS(9)
CURRENT ADDRESS(9)
EN
512
221

FIG. 6

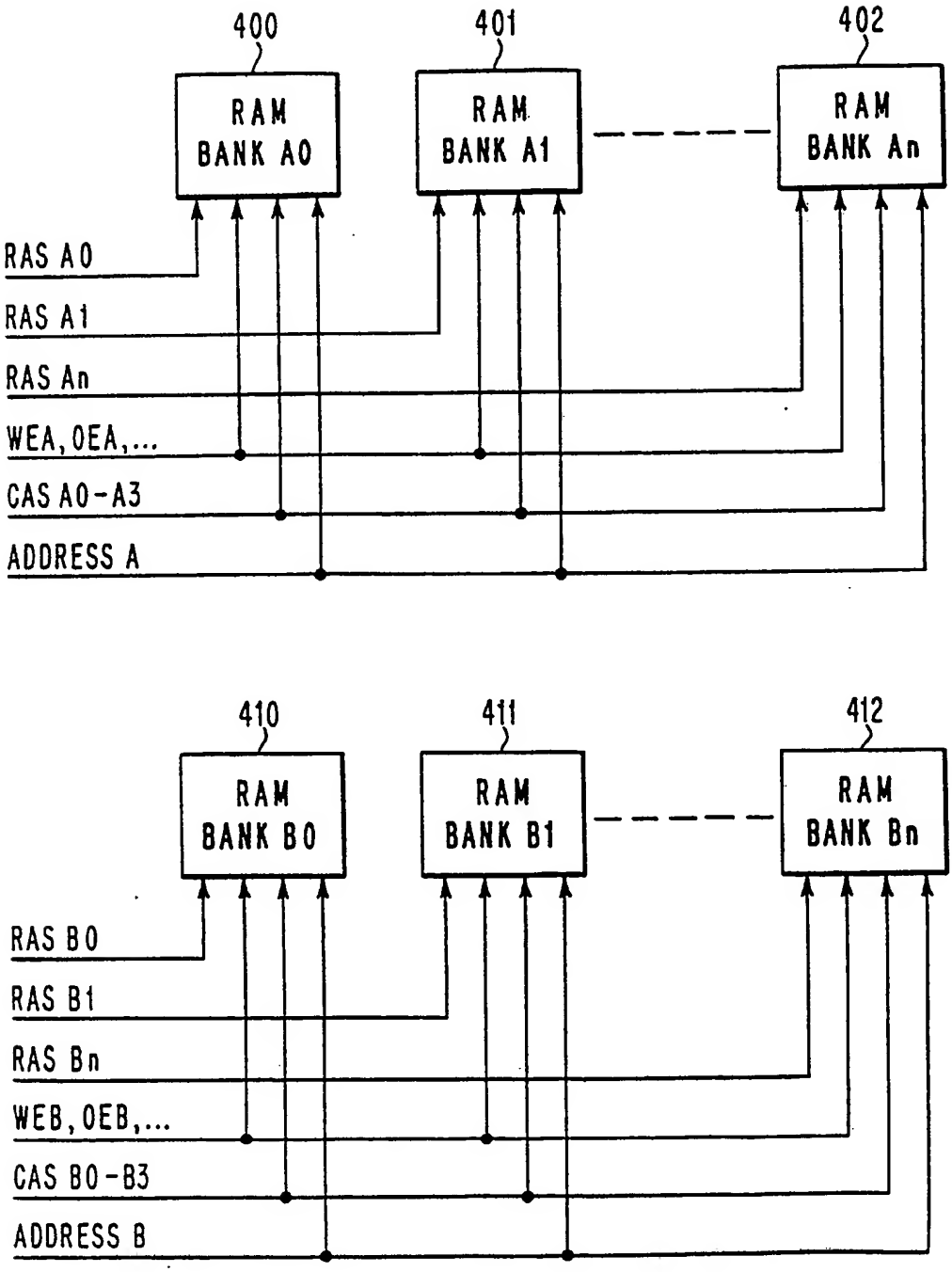

400
401
402
RAM BANK A0
RAM BANK A1
RAM BANK An
RAS A0
RAS A1
RAS An
WEA, OEA,...
CAS A0-A3
ADDRESS A
410
411
412
RAM BANK B0
RAM BANK B1
RAM BANK Bn
RAS B0
RAS B1
RAS Bn
WEB, OEB,...
CAS B0-B3
ADDRESS B

FIG. 7

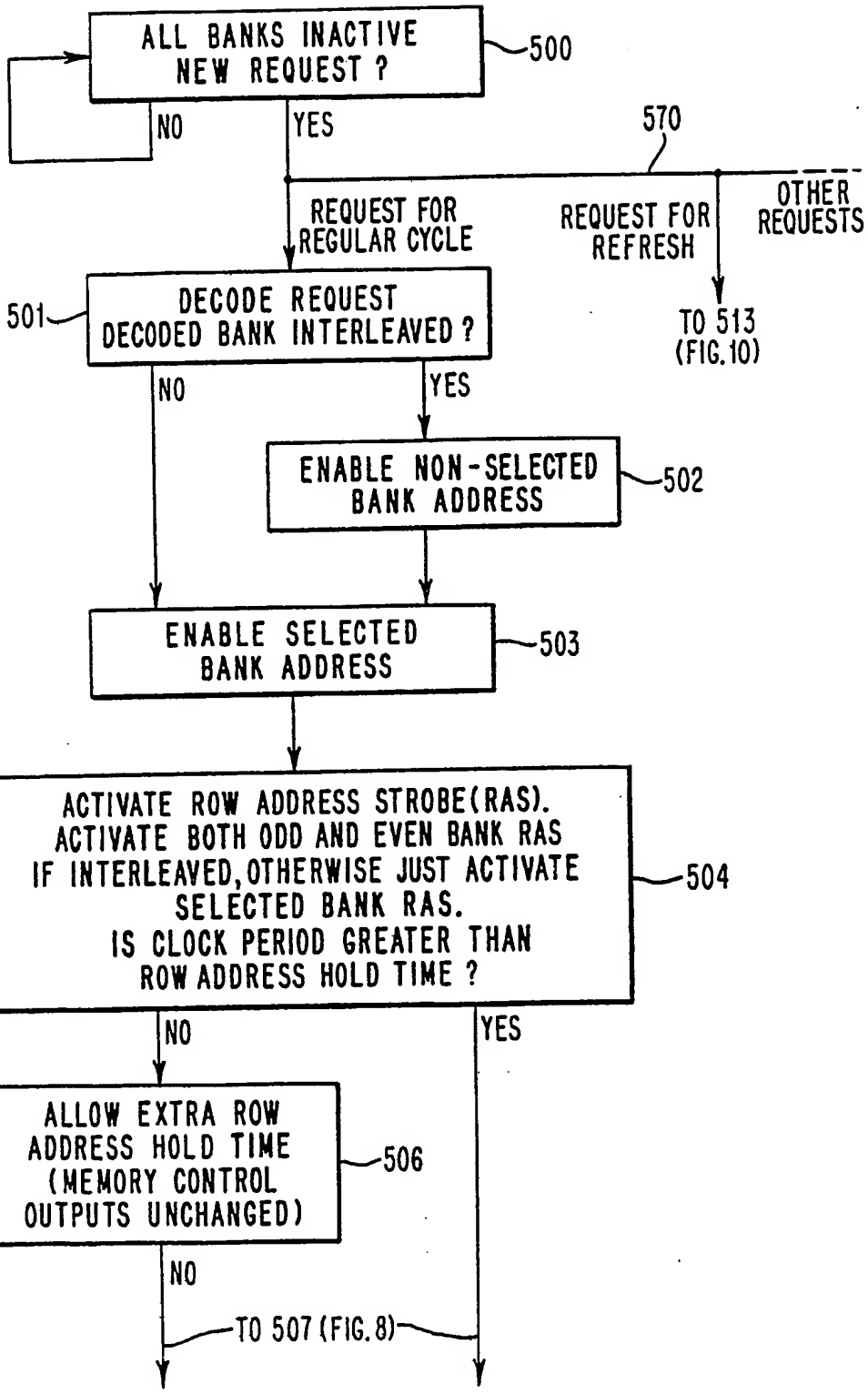

ALL BANKS INACTIVE NEW REQUEST ?
500
NO
YES
570
REQUEST FOR REGULAR CYCLE
REQUEST FOR REFRESH
OTHER REQUESTS
TO 513 (FIG. 10)
501
DECODE REQUEST DECODED BANK INTERLEAVED ?
NO
YES
ENABLE NON-SELECTED BANK ADDRESS
502
ENABLE SELECTED BANK ADDRESS
503
ACTIVATE ROW ADDRESS STROBE(RAS). ACTIVATE BOTH ODD AND EVEN BANK RAS IF INTERLEAVED, OTHERWISE JUST ACTIVATE SELECTED BANK RAS. IS CLOCK PERIOD GREATER THAN ROW ADDRESS HOLD TIME ?
504
NO
YES
ALLOW EXTRA ROW ADDRESS HOLD TIME (MEMORY CONTROL OUTPUTS UNCHANGED)
506
NO
TO 507 (FIG. 8)

FIG. 8

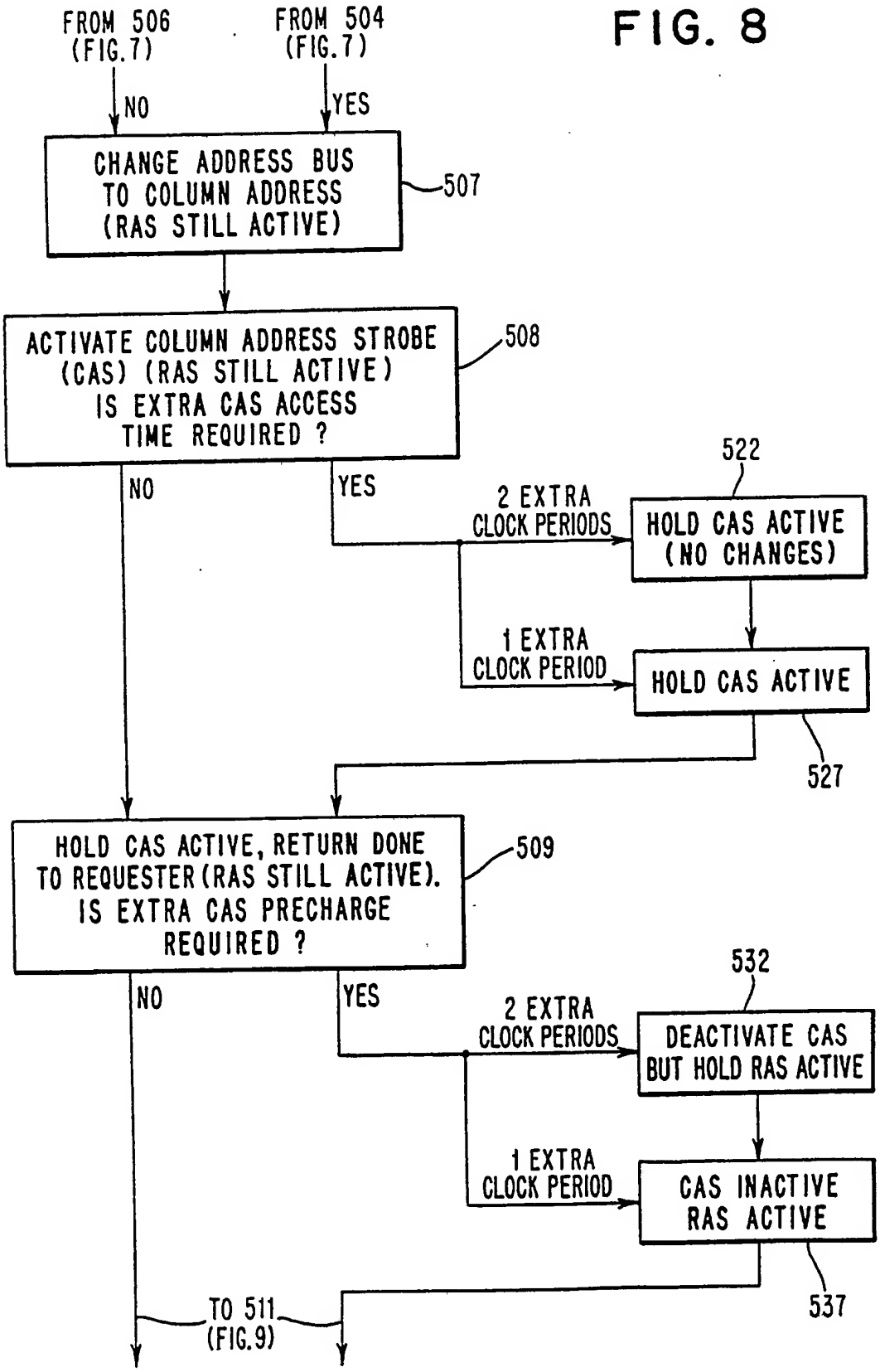

FROM 506 (FIG.7)
FROM 504 (FIG.7)
NO
YES
CHANGE ADDRESS BUS TO COLUMN ADDRESS (RAS STILL ACTIVE)
507
ACTIVATE COLUMN ADDRESS STROBE (CAS) (RAS STILL ACTIVE) IS EXTRA CAS ACCESS TIME REQUIRED ?
508
NO
YES
2 EXTRA CLOCK PERIODS
522
HOLD CAS ACTIVE (NO CHANGES)
1 EXTRA CLOCK PERIOD
HOLD CAS ACTIVE
527
HOLD CAS ACTIVE, RETURN DONE TO REQUESTER (RAS STILL ACTIVE). IS EXTRA CAS PRECHARGE REQUIRED ?
509
NO
YES
2 EXTRA CLOCK PERIODS
532
DEACTIVATE CAS BUT HOLD RAS ACTIVE
1 EXTRA CLOCK PERIOD
CAS INACTIVE RAS ACTIVE
537
TO 511 (FIG.9)

FIG. 9

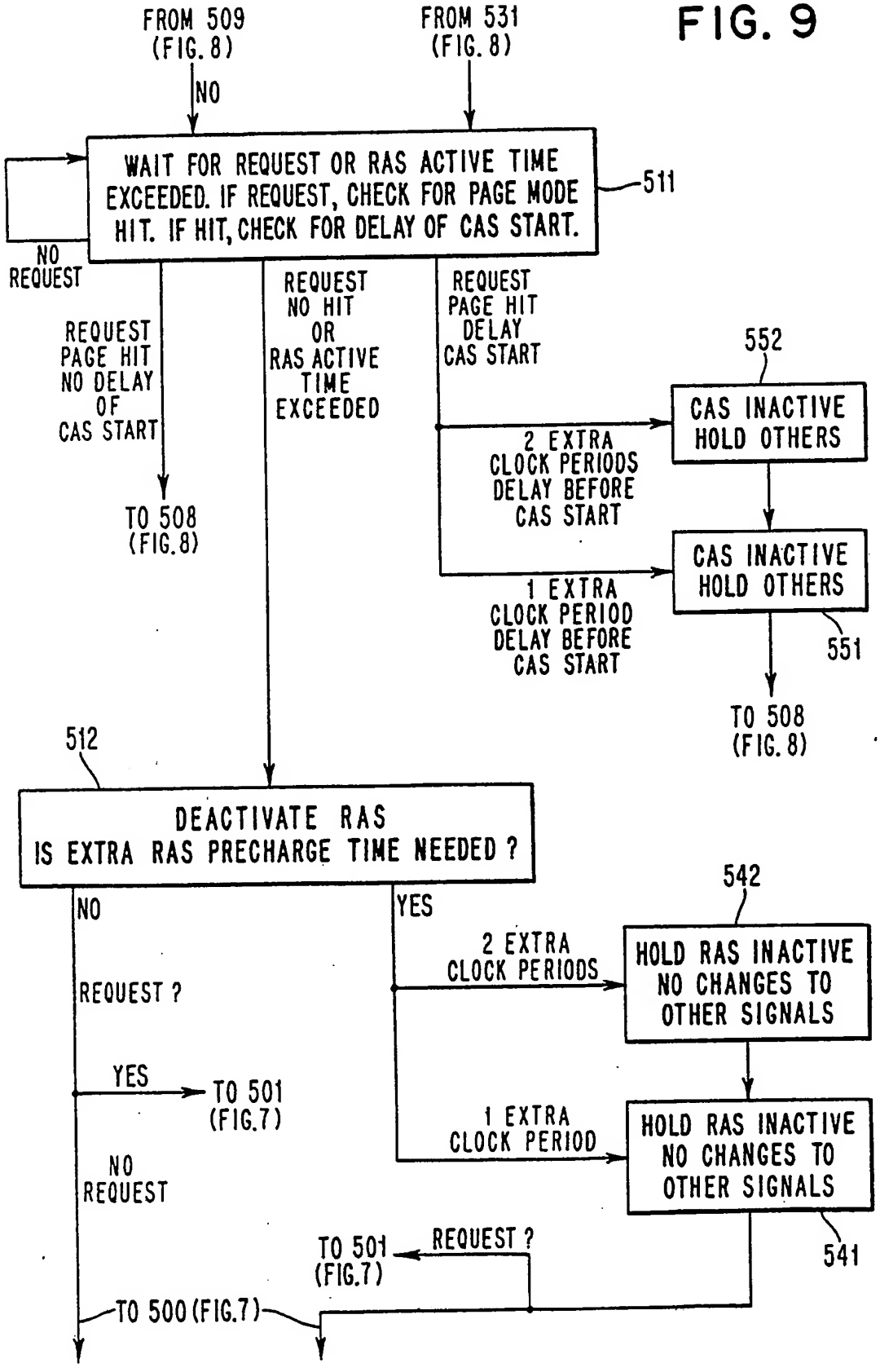
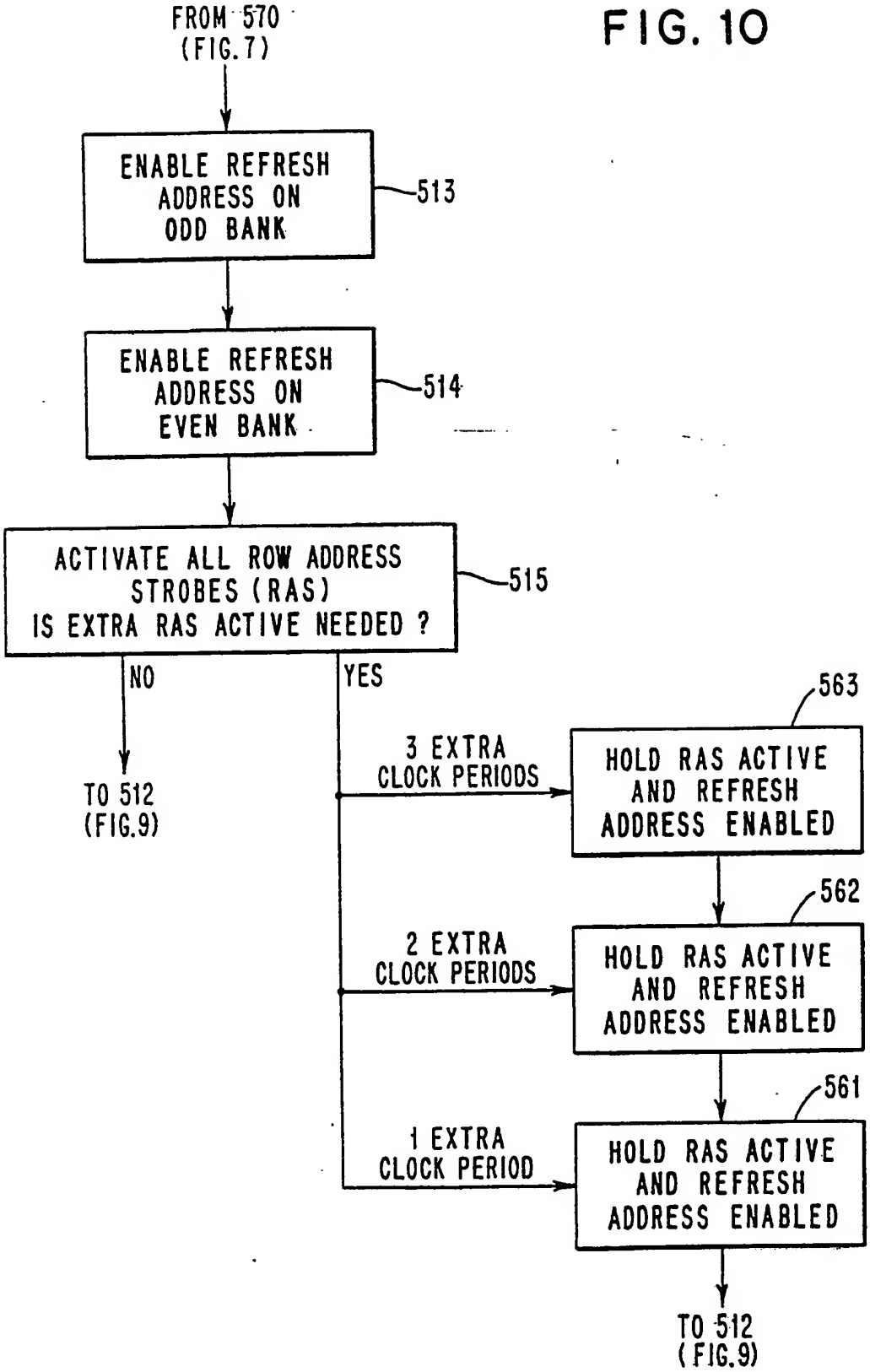

FROM 509 (FIG. 8)
NO
FROM 531 (FIG. 8)
WAIT FOR REQUEST OR RAS ACTIVE TIME EXCEEDED. IF REQUEST, CHECK FOR PAGE MODE HIT. IF HIT, CHECK FOR DELAY OF CAS START.
511
NO REQUEST
REQUEST PAGE HIT NO DELAY OF CAS START
TO 508 (FIG. 8)
REQUEST NO HIT OR RAS ACTIVE TIME EXCEEDED
REQUEST PAGE HIT DELAY CAS START
552
CAS INACTIVE HOLD OTHERS
2 EXTRA CLOCK PERIODS DELAY BEFORE CAS START
CAS INACTIVE HOLD OTHERS
1 EXTRA CLOCK PERIOD DELAY BEFORE CAS START
551
TO 508 (FIG. 8)
512
DEACTIVATE RAS
IS EXTRA RAS PRECHARGE TIME NEEDED ?
NO
REQUEST ?
YES
TO 501 (FIG. 7)
NO REQUEST
YES
542
2 EXTRA CLOCK PERIODS
HOLD RAS INACTIVE NO CHANGES TO OTHER SIGNALS
1 EXTRA CLOCK PERIOD
HOLD RAS INACTIVE NO CHANGES TO OTHER SIGNALS
541
REQUEST ?
TO 501 (FIG. 7)
TO 500 (FIG. 7)

FIG. 10


FROM 570
(FIG.7)
ENABLE REFRESH ADDRESS ON ODD BANK
513
ENABLE REFRESH ADDRESS ON EVEN BANK
514
ACTIVATE ALL ROW ADDRESS STROBES (RAS)
IS EXTRA RAS ACTIVE NEEDED ?
515
NO
TO 512
(FIG.9)
YES
3 EXTRA CLOCK PERIODS
563
HOLD RAS ACTIVE AND REFRESH ADDRESS ENABLED
2 EXTRA CLOCK PERIODS
562
HOLD RAS ACTIVE AND REFRESH ADDRESS ENABLED
1 EXTRA CLOCK PERIOD
561
HOLD RAS ACTIVE AND REFRESH ADDRESS ENABLED
TO 512
(FIG.9)